

Title: Space-Time Methods for Hyperbolic Conservations
Laws

Author(s): Robert B. Lowrie
Philip L. Roe
Bram Van Leer

Submitted to: Proceedings of ICASE/LaRC Workshop on Barriers and
Challenges in Computational Fluid Dynamics
August 5-7, 1996
Langley Research Center
Hampton, Virginia

Los Alamos
National Laboratory

Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by the University of California for the U.S. Department of Energy under contract W-7405-ENG-36. By acceptance of this article, the publisher recognizes that the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy. The Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

SPACE-TIME METHODS FOR HYPERBOLIC CONSERVATION LAWS

ROBERT B. LOWRIE

Los Alamos National Laboratory

Scientific Computing Group

AND

PHILIP L. ROE AND BRAM VAN LEER

University of Michigan

Aerospace Engineering Department

1. Introduction

Two major challenges for computational fluid dynamics are problems that involve wave propagation over long times and problems with a wide range of amplitude scales. An example with both of these characteristics is the propagation *and* generation of acoustic waves, where the mean-flow amplitude scales are typically orders-of-magnitude larger than those of the generated acoustics. Other examples include vortex evolution and the direct simulation of turbulence. All of these problems require greater than second-order accuracy, whereas for nonlinear equations, most current methods are at best second-order accurate. Of the higher-order (greater than second-order) methods that do exist, most are tailored to high-spatial resolution, coupled with time integrators that are only second or third-order accurate. But for wave phenomena, time accuracy is as important as spatial accuracy.

One property of successful second-order methods is that they attempt to be faithful to the physics of hyperbolic problems. To develop higher-order methods, particularly for unsteady problems, it is tempting to violate this philosophy. Typically, higher accuracy is obtained by increasing the size of the update stencil. Instead, our aim is to develop time-accurate methods that minimize the size of the update stencil.

The approach in this study is strongly motivated by the physics of hyperbolic conservation laws. Specifically, we insist that a numerical method's discrete zone of dependence should only be slightly larger (for stability) than the physical zone of dependence. Consider a two-time-level method with a Courant number less than 1. In one dimension, only three cells should contribute to a cell's update; the cell itself and its immediate neighbors. We will refer to such methods as *compact*.

Compact methods have the following potential benefits:

- The methods are explicit.
- Parallel implementations using domain decomposition only need a single buffer cell at domain boundaries, which is updated only once per time step.
- Boundary procedures are straightforward.
- The effects of discontinuities are localized.

Additional benefits will also be realized, but to discuss them will first require some analysis.

Nearly all of the popular higher-order methods are not compact. In particular, ENO-based methods, and methods based on multi-stage time integration, are not compact. For a higher-order method to be compact, either more time levels must be included in the update, or more data must be carried in each cell. The choice in this study is to carry more data in each cell, with a two-time-level approach.

Methods that carry more data in each cell, than simply a cell-average or point-value, are not a new idea. In fact, the approach outlined here is a generalization of Van Leer's Scheme III (van Leer, 1977), which carries the solution average and first moment (or derivative) in each cell. Scheme III is third-order accurate in space and time.¹ This scheme can also be derived in a finite-element context (Johnson and Pitkaranta, 1986), and is referred to as the Discontinuous Galerkin (DG) method. It is the possibility of deriving schemes with improved order-of-accuracy that makes this approach potentially superior to using the extra storage for mesh refinement.

For a bibliography of other work using DG, see (Lowrie, 1996). A notable multi-stage implementation is given by (Atkins and Shu, 1996), where efficiency is gained by eliminating the need for quadrature. The multi-stage DG method may be ideal for steady-state problems; however, no multi-stage scheme can be compact. Nevertheless, since each *stage* of the method is compact, it has many of the advantages of compact schemes that are outlined above.

The DG method in this study follows the 'space-time' approach. Control volumes (elements) are defined in space and time, and then a polynomial representation (which includes the time variable) of the solution is found in each element. The elements are arranged so that when solving a linear equation, each element can be solved for explicitly in a marching procedure. For nonlinear equations, the method is *point-implicit*, in that the solution in each element requires the solution of a small system of equations, and only a small number of elements may be coupled. Moreover, the implicitness is weak, arising solely from the nonlinearity, so that rapid iteration is possible.

The approach taken in this study is by no means simple, and at this time the computational cost is high.² However, the method has a strong theoretical foundation and many of its properties are highly desirable. Future work should make the present method, or related methods, practical for a broader range of problems.

2. Conservation Laws

Consider a conservation law of m -equations in d -space dimensions, written as

$$\partial_t \mathbf{u} + \nabla \cdot \mathbf{f}(\mathbf{u}) = \mathbf{0}, \quad (1)$$

where \mathbf{u} is the vector of conservation variables, $\nabla \equiv (\partial_{x_1}, \dots, \partial_{x_d})$, and $\mathbf{f} \equiv (\mathbf{f}_1, \dots, \mathbf{f}_d)$. Another form that will be used in this study is

$$\vec{\nabla} \cdot \vec{\mathbf{f}} = \mathbf{0}, \quad (2)$$

where $\vec{\nabla} \equiv (\nabla, \partial_t)$ and $\vec{\mathbf{f}} \equiv (\mathbf{f}, \mathbf{u})$. The notation is that \underline{v} is a vector in space, while \vec{v} is a space-time vector.

¹In a norm that measures the error in the least-damped mode of the solution.

²Note that the "quadrature-free" idea of (Atkins and Shu, 1996) has not yet been implemented, which may increase performance substantially.

Define the matrix

$$\mathbf{A}_\ell \equiv \underline{\ell} \cdot \underline{\mathbf{A}}, \quad (3)$$

with $\underline{\ell}$ a unit vector, and each component of $\underline{\mathbf{A}}$ is the $m \times m$ Jacobian matrix corresponding to $\underline{\mathbf{f}}$. We assume (1) is hyperbolic; that is, for every $\underline{\ell}$, \mathbf{A}_ℓ has real eigenvalues, $\lambda_{\ell,k}$, $k = 1, 2, \dots, m$, and distinct eigenvectors. The Euler equations of gas dynamics are an example of a hyperbolic system.

3. Discontinuous Galerkin

3.1. FORMULATION

Let the solution domain Ω be divided into a set of N_e non-overlapping control volumes (elements), $\{\Omega_e\}$. Each Ω_e is allowed to be any type of polygon, with boundary $\partial\Omega_e$. A sample space-time mesh for $d = 1$, between two time levels, is shown in Figure 1.

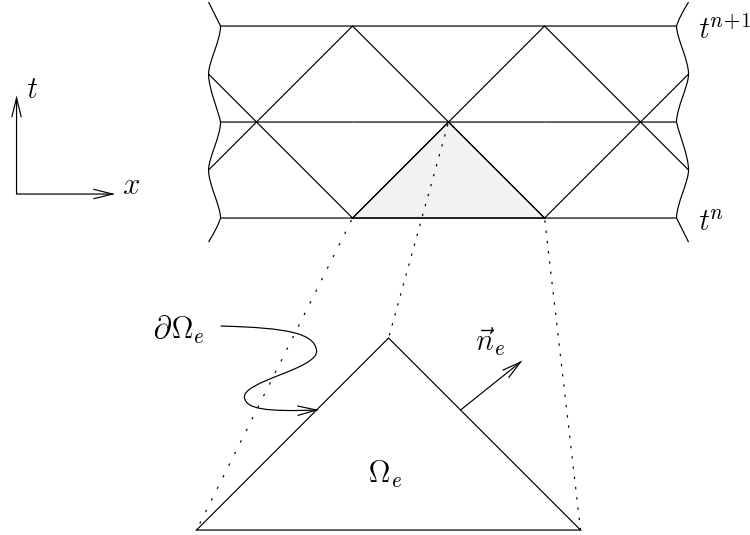


Figure 1. Sample Space-Time Mesh and Element Definition.

The solution in each element is written in terms of the parameter vector, $\mathbf{w}(\mathbf{u})$ (Roe, 1981). For many conservation laws, the quantities \mathbf{u} and \mathbf{f}_i can be written as quadratic functions of the components of \mathbf{w} . This property is used to allow exact evaluation of the flux integrals.

The numerical solution sought will be continuous within each Ω_e , but possibly discontinuous across element boundaries. The solution within each element- e will be referred to as $\mathbf{w}_e(\vec{x})$, $\forall \vec{x} \in \Omega_e$, and on the element boundary by $\mathbf{w}_b(\vec{x})$, $\forall \vec{x} \in \partial\Omega_e$. Note that in general

$$\mathbf{w}_e(\vec{x}) \neq \mathbf{w}_b(\vec{x}), \quad \forall \vec{x} \in \partial\Omega_e.$$

Indeed, the boundary value may be equal to one of the two neighboring element values, or some combination thereof. The precise definition of the boundary value will be given in the next section.

In weak form, (2) may be written for each element as

$$\oint_{\partial\Omega_e} \Phi \vec{f}_b \cdot \vec{n}_e dS - \int_{\Omega_e} \vec{f}_e \cdot \vec{\nabla} \Phi dV = 0, \quad (4)$$

where \vec{n}_e is the outward boundary unit normal, and $\Phi = \Phi(\vec{x})$ is a suitable test function, to be defined later. Note that Equation (4) can be related to the finite-volume approach by taking $\Phi(\vec{x}) = 1$.

3.2. FACE DEFINITIONS

In this section, the element-boundary values, $(\cdot)_b$, are defined. Let $\partial\omega_e$ be an element face that separates two elements Ω_e and Ω_{e^*} . Borrowing from the finite-volume approach, the interface value is written as a function of the values in the adjacent elements;

$$\mathbf{w}_b(\vec{x}) = \mathcal{F}(\mathbf{w}_e(\vec{x}), \mathbf{w}_{e^*}(\vec{x})), \quad \forall \vec{x} \in \partial\omega_e.$$

The function \mathcal{F} will depend on the face “type.” Our present code permits two types of

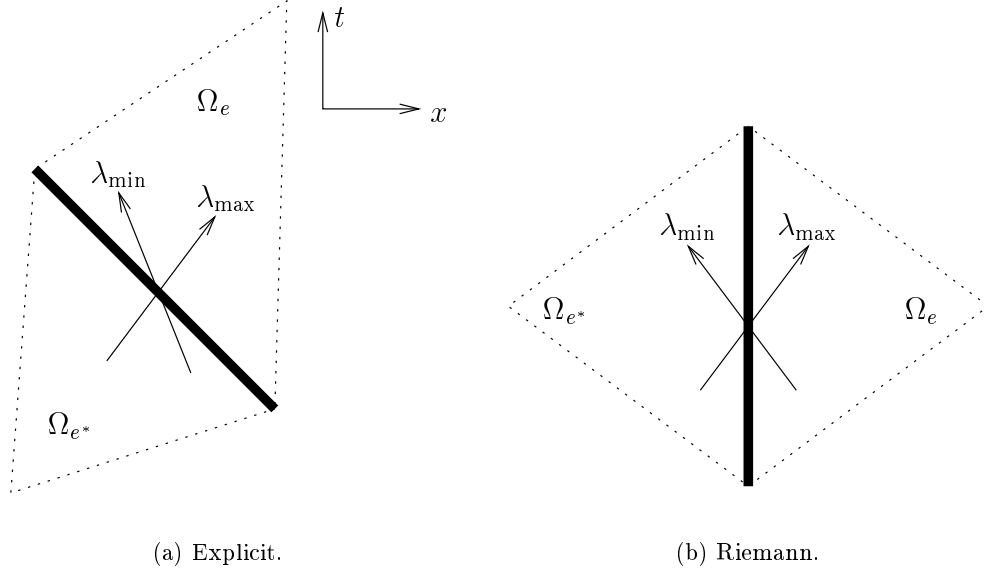


Figure 2. Face Definitions for $d = 1$. λ_{\min} and λ_{\max} correspond to the local minimum and maximum eigenvalues, respectively.

faces (refer also to Figure 2):

1. *Riemann Face*: A face that is aligned with the t -axis ($n_{e,t} = 0$). On such faces, a (approximate) Riemann solver is used.
2. *Explicit Face*: The face orientation is such that all of the characteristic paths cross the face in the same direction; that is, the quantity

$$\mu = (\underline{\ell}, \lambda_{\ell,k}) \cdot \vec{n}_e,$$

is either strictly positive (also referred to as an “Outflow Face”), or strictly negative (“Inflow Face”), for all k and $\underline{\ell}$, and all $\vec{x} \in \partial\omega_e$. The vector $(\underline{\ell}, \lambda_{\ell,k})$ is defined via (3). In practice, the calculation of $\lambda_{\ell,k}$ for all $\underline{\ell}$ and $\vec{x} \in \partial\omega_e$ is not needed, as long as a reasonable local value is used, along with a safety factor. The boundary value is then set as

$$\mathbf{w}_b(\vec{x}) = \begin{cases} \mathbf{w}_{e^*}(\vec{x}) & \text{if } \mu < 0 : \quad \text{“Inflow”} \\ \mathbf{w}_e(\vec{x}) & \text{if } \mu > 0 : \quad \text{“Outflow”} \end{cases} \quad (5)$$

for all $\vec{x} \in \partial\omega_e$. Note that $\mu \equiv 0$ is only permissible for Riemann Faces.

The above definitions are what give the scheme an “upwind” character. In fact, an Explicit Face can be thought of as a Riemann Face on which the Riemann solution is known *a priori*. Also, in a region where all of the eigenvalues have the same sign, a Riemann Face satisfies the definition of an Explicit Face.

4. Implementation

4.1. SPACE-TIME MESHES

This section defines the space-time meshes on which the discrete form of the conservation law will be solved. How the mesh is defined will greatly influence the cost of the DG method. The underlying principle will be to form each space-time mesh in such a way that the numerical method is at worst point implicit.

4.1.1. 1-D Meshes

In 1-D, two space-time meshes are used. The Riemann Mesh, shown in Figure 3a, contains Riemann Faces which couple pairs of elements implicitly. To avoid Riemann Faces altogether, the Staggered Mesh in Figure 3b will also be used. For both meshes, the diagonal Explicit Faces satisfy Equation (5) as long as the Courant number is less than 1. This condition is in fact the stability constraint.

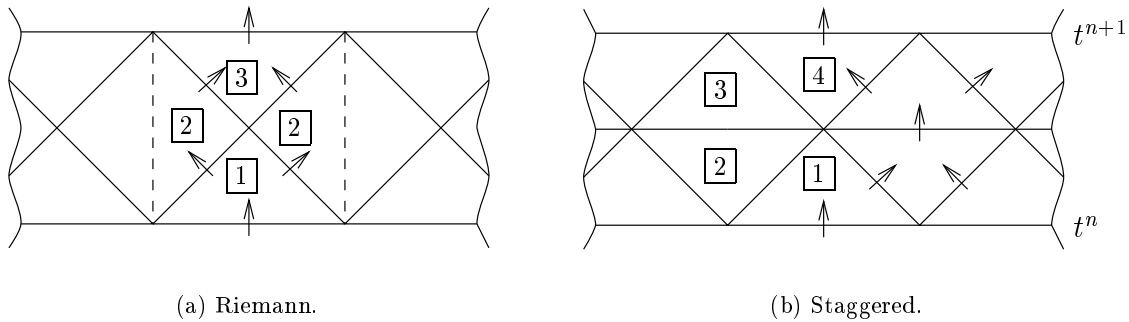


Figure 3. Various Space-Time Mesh Segments. Solid lines denote Explicit Faces, with arrows indicating the flow of information. Dashed lines are Riemann Faces. The element types are numbered according to the order in which they are solved.

Figure 3 also indicates the order in which the elements are solved. The solution procedure for the Riemann Mesh is as follows:

1. Solve in each of the elements of family [1]. The flux on the bottom face is either from the initial condition, or the solution in element [3] from the previous time level.
2. Solve in the elements of family [2]. In general, these elements are coupled in pairs by the Riemann Faces. The two elements are not coupled implicitly if all of the eigenvalues are of the same sign.
3. Solve in each of the elements of family [3].
4. Proceed to the next time level.

The Staggered Mesh is solved in a similar fashion, the only difference being that no elements are coupled implicitly. For the Euler equations, the computational cost of using the Riemann Mesh is approximately 70% more than the Staggered Mesh.

A mesh can also be defined that is made up of segments of the Riemann and Staggered Meshes. In particular, to enforce boundary conditions (other than periodic) the Staggered Mesh uses a Riemann Mesh segment at the boundary.

4.1.2. 2-D Meshes

The 2-D space-time mesh is an extension of the 1-D Staggered Mesh of Figure 3b. The underlying 2-D spatial mesh is a quadrilateral mesh. To visualize the space-time mesh, Figure 4 shows the order in which elements are solved, over $\frac{1}{2}\Delta t$. In the next half-time step, the mesh is staggered; the Step (1) pyramid base, in the $\frac{1}{2}\Delta t < t \leq \Delta t$ interval, is coincident with the Step (3) pyramid base of the $0 < t \leq \frac{1}{2}\Delta t$ interval. Note that at each step, none of the element solutions are coupled, just as in 1-D Staggered Mesh. For Step (2), the same final solution results if the y -axis “valley” elements are solved before the x -axis “valley” elements. The method does not use “operator splitting.” As a consequence, the results show that this mesh does not exhibit a loss of accuracy when the advection direction is skewed with respect to the spatial mesh.

A 2-D analogy to the 1-D Riemann Mesh also exists, along with meshes based on a 2-D triangular spatial mesh and a 3-D hexahedral mesh; see (Lowrie, 1996). Only the Staggered Mesh has been used for 2-D problems, with Riemann Faces used on the domain boundaries to enforce boundary conditions.

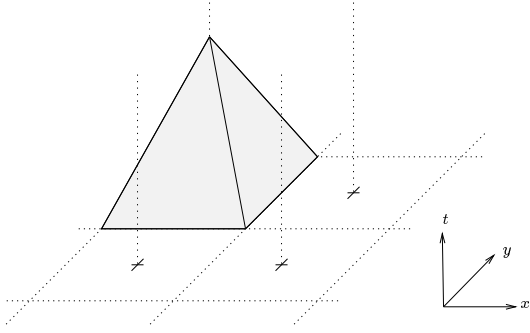
4.2. ALGEBRAIC SYSTEM

To solve Equation (4) numerically, in each element the solution is approximated as

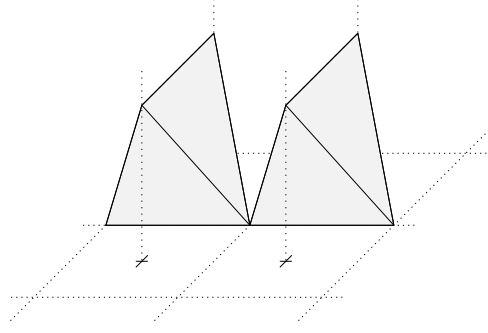
$$\mathbf{w}_e(\vec{x}) = \sum_{j=1}^N \phi_{e,j}(\vec{x}) \mathbf{c}_{e,j}, \quad \forall \vec{x} \in \Omega_e,$$

where the $\{\phi_{e,j}\}$, $j = 1, 2, \dots, N$ are polynomial basis functions over the element Ω_e . DG(k) will refer to the DG method with each $\phi_{e,j} \in \mathcal{P}_k(\Omega_e)$, where $\mathcal{P}_k(\Omega_e)$ is the space of polynomials of maximum order k defined on Ω_e . The variable $\mathbf{w} = \mathbf{w}(\mathbf{u})$ is the parameter vector. There are N -unknown m -vectors in each element, namely the expansion coefficient vectors, $\mathbf{c}_{e,j}$.

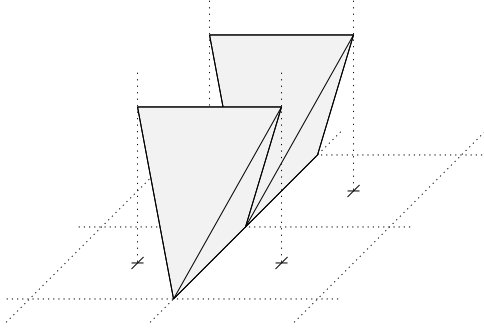
To generate the necessary N vector equations, the Galerkin approach is to choose $\Phi = \phi_{e,i}$, $i = 1, 2, \dots, N$, in Equation (4). This choice results in the minimization property described in (Lowrie, 1996). For a nonlinear conservation law, the resulting system of equations is nonlinear in the $\mathbf{c}_{e,j}$. To solve this system, a Newton-Kantorovich approach is taken.



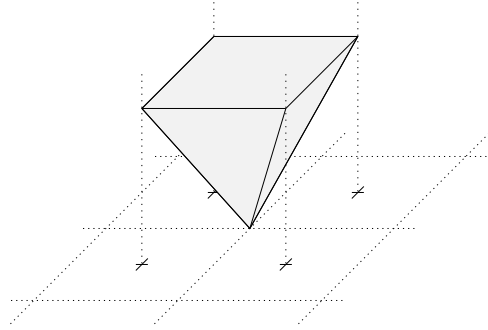
Step (1): Solve in the pyramid elements covering each x, y mesh cell.



Step (2): Solve in the tetrahedral elements that fill the x -axis valleys of the pyramid elements in Step [1].



Step (2)-continued: Solve in the tetrahedral elements that fill the y -axis valleys of the pyramid elements in Step [1].



Step (3): Solve in the inverted-pyramid elements.

Figure 4. Element Solution Order for the 2-D Staggered Mesh. Only $\frac{1}{2}\Delta t$ of mesh is shown.

Consider the general case where the solution of N_c elements are coupled. Linearizing (4) gives a system of equations of the form

$$\mathbf{M}(\delta \mathbf{c}) = -\mathbf{r}, \quad (6)$$

where $\delta \mathbf{c}$ is the update vector to the expansion coefficients, and \mathbf{M} is a $N_c \times N_c$ block matrix. Each submatrix block is an $(mN) \times (mN)$ matrix. From the boundary definitions, the coupling of elements can only occur across Riemann Faces. Therefore, the maximum $N_c = 2$ for the 1-D Riemann Mesh, and $N_c = 1$ for the Staggered Meshes.

The integrals are computed numerically using Gaussian quadrature. Note that if Equation (1) is a linear system, then \mathbf{M} is a constant matrix, which can be inverted once and stored for each class of coupled elements. For nonlinear systems, the cost can be reduced without a significant decrease in accuracy by performing only a single iteration of (6), and

by computing \mathbf{M} using the DG(0) solution (which can often be found explicitly). This eliminates the need for quadrature when computing \mathbf{M} .

Boundary conditions are applied by defining a “ghost” Gauss point at each physical Gauss point that is on the boundary. A Riemann problem is then solved to determine the boundary flux. The boundary procedure is the same for any order-of-accuracy, unlike methods that require special difference stencils.

5. Accuracy

The DG method has the following properties:

1. Conservation.
2. Stability, and satisfies an entropy condition, for Courant numbers less than 1 and any order-of-accuracy.
3. A minimization property.
4. High accuracy.

Each of these properties are discussed in (Lowrie, 1996). A brief overview of the accuracy will be given in this section.

For a method using an order- k interpolant, the expectation is that at best the error will converge as $\mathcal{O}(h^{k+1})$, where h is some measure of the mesh size. However, a Fourier analysis shows that DG(k) converges as $\mathcal{O}(h^{2k+1})$ in a certain norm, indicating that DG(k) has a *superconvergence* property. This norm, denoted by L_p^{ev} , measures only the error in the evolution of the initial condition projected onto the accurate mode of the update operator.³ The L_p^{ev} -norm will be related to more standard norms in the remainder of this section.

For a calculation to any fixed time $t > 0$, in the limit $h \rightarrow 0$, all of the spurious modes will be damped out. Therefore, any other norm will include a term proportional to the evolution error, plus the particular norm’s measure of the initial condition projected onto the accurate mode (a one-time contribution).

A norm typically used to measure the solution accuracy at a given time-level is

$$L_p(v) = \left\{ \frac{1}{|\Omega_d|} \int_{\Omega_d} |v(\underline{x}) - v_{\text{exact}}(\underline{x})|^p dx \right\}^{1/p},$$

where Ω_d is the spatial domain, and v, v_{exact} are the numerical and exact solutions of a representative variable of the conservation law. A finite-element analysis shows that DG(k) converges in L_2 as $\mathcal{O}(h^{k+1/2})$ (Jaffre *et al.*, 1995). However, for many smooth solutions on ‘regular’ meshes, practitioners often realize $\mathcal{O}(h^{k+1})$ accuracy (Jaffre *et al.*, 1995; Richter, 1988). By including the initial-projection error, the Fourier analysis gives a convergence rate of $\mathcal{O}(h^{k+1})$ in L_p .

Another norm studied here is the error in the cell averages, denoted by $\bar{L}_p(v)$. The Fourier analysis for this norm gives $\mathcal{O}(h^{k+2})$ for $k > 0$, and $\mathcal{O}(h)$ for $k = 0$. Although L_p and \bar{L}_p have slower convergence than L_p^{ev} , the consequences of the superconvergence property will be apparent in the results of Section 6.1.

Note that the semi-discrete version of DG, with $(k+1)$ -multi-stage, is $\mathcal{O}(h^{k+1})$ accurate (Cockburn and Shu, 1989), and therefore does not have the superconvergence property.

³The accurate mode is the least-damped mode. There are k -spurious modes.

However, (Atkins and Shu, 1996) later observed $\mathcal{O}(h^3)$ convergence in the cell-averages for $k = 1$, using a 3-stage time integration. Since the 3-stage method is a third-order time integrator, obviously there is some mode that is advected with third-order accuracy. What Atkins and Shu have apparently shown is that the projection of the initial condition onto this mode is also third-order.

6. Results

6.1. SCALAR ADVECTION

Results are now presented for the linear equation

$$\partial_t u + \partial_x u = 0,$$

on $0 \leq x \leq 1$, with the initial condition $u(x, 0) = \sin(2\pi x)$, and periodic boundary conditions. Figure 5 shows the order-of-accuracy history for the Staggered Mesh, $\nu = 0.8$, using the L_1 and \bar{L}_1 -norms. Similar results are obtained for the Riemann Mesh. At least for early times, $DG(k)$ follows the Fourier analysis' prediction of $\mathcal{O}(h^{k+1})$ in L_p . An interesting phenomenon can occur, however, as shown by the results for $DG(1)$. Given enough time, the evolution error will accumulate, and overcome the initial-projection error, denoted as L_1^0 . Past this time, for a given stage in mesh refinement, the accuracy convergence is dictated by the evolution error, L_1^{ev} , which converges as $\mathcal{O}(h^{2k+1})$.

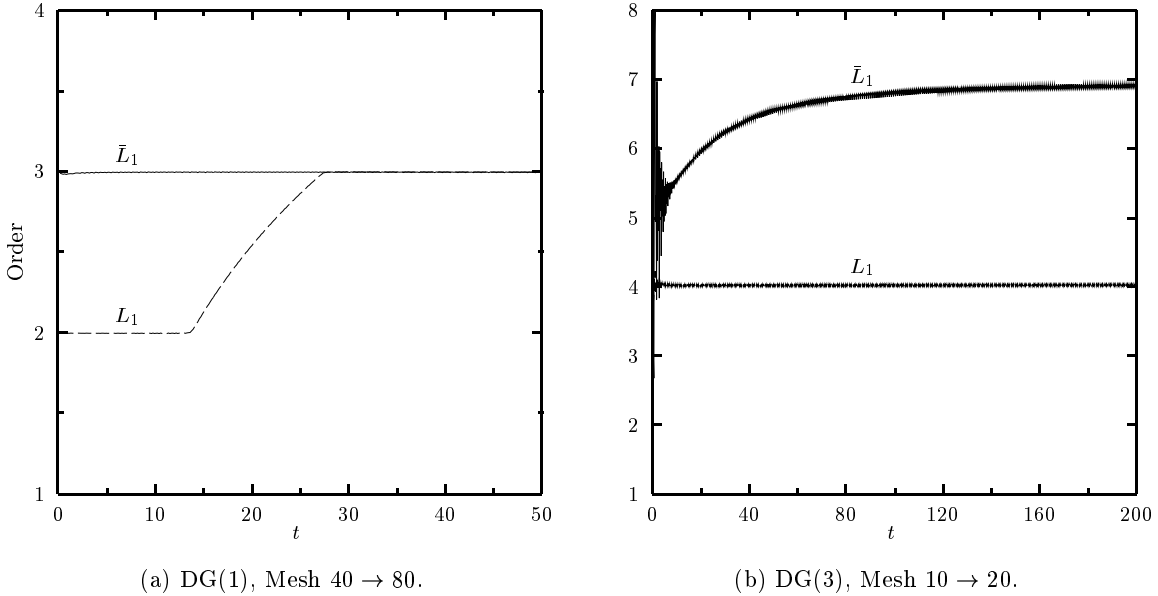


Figure 5. Order-of-Accuracy History, Sine Wave, Staggered Mesh, $\nu = 0.8$, using L_1 and \bar{L}_1 -norms. The two mesh sizes used for the order calculation are indicated.

Such behavior is also evident in the \bar{L}_1 -norm for $k = 3$. Since the initial error $\bar{L}_1^0 \ll L_1^0$, the order asymptotes to apparent $\mathcal{O}(h^{2k+1})$ -accuracy much more quickly in \bar{L}_1 than L_1 . The oscillations in the order are the result of the spurious modes. Note that the time at

which the evolution error overcomes the initial-projection error increases with mesh size, and therefore strictly speaking the convergence in \bar{L}_1 is $\mathcal{O}(h^{k+2})$. However, the finest mesh used in each case here gives results that are well-resolved. At the final time shown in each plot, DG(1) has an error given by $\log_{10}(\bar{L}_1) = -3.56$, while DG(3)'s error is -7.51. For many practical problems where long-time integration is required, there is the possibility of realizing $\mathcal{O}(h^{2k+1})$ -accuracy, particularly in \bar{L}_p .

For smooth solutions, nonlinearity does not seem to destroy the superconvergence property, but the time-histories are more complicated. See (Lowrie, 1996) for results when solving the Euler equations.

6.2. SHU-OSHER PROBLEM

The problem of (Shu and Osher, 1989) corresponds to a Mach 3 shock propagating into a sinusoidal density wave. This problem requires a limiter; see (Lowrie, 1996) for specifics on the limiter used. Figure 6 shows the DG(1) cell-average densities, using 400 cells on the Staggered Mesh. The “exact” solution here is the 1600-cell solution. For this problem, the Riemann Mesh solution is very similar. Also, the 800-cell solution is nearly indistinguishable from the exact solution. The results here compare favorably with the TVD results of (Huynh, 1995; Cockburn *et al.*, 1989).

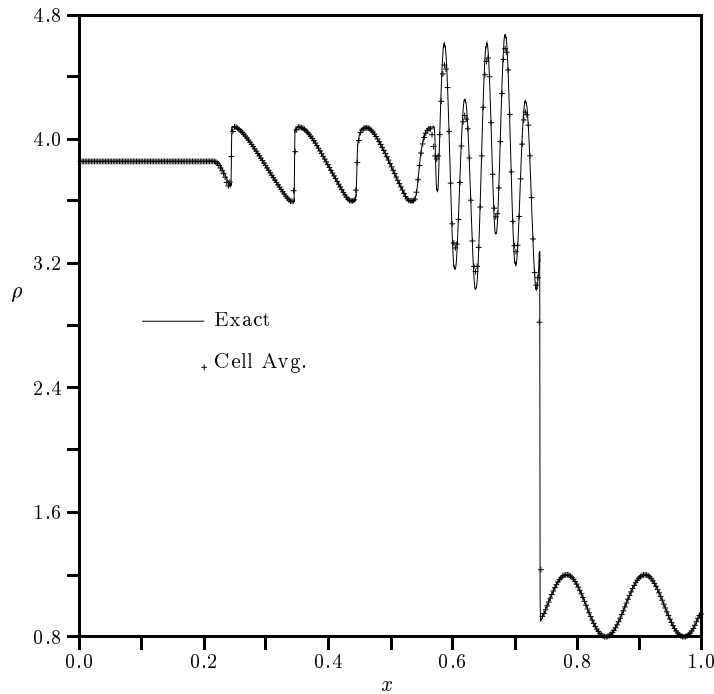


Figure 6. Shu-Osher Problem, DG(1), Staggered Mesh, $\nu = 0.8$, 400 cells, $t = 0.18$.

6.3. PRESSURE PULSE IN FREESTREAM OVER WALL

This problem has an initial condition of a Gaussian-pressure pulse above a plane wall, immersed in a steady flow. This is the “Category IV” problem described in the (NAS,

1994) workshop proceedings. The Workshop participants used a 200×200 mesh for this problem. Figure 7 shows a solution of the full Euler equations using DG(3) on a 25×25 mesh.⁴ On such a coarse mesh, the initial Gaussian is spread over approximately three

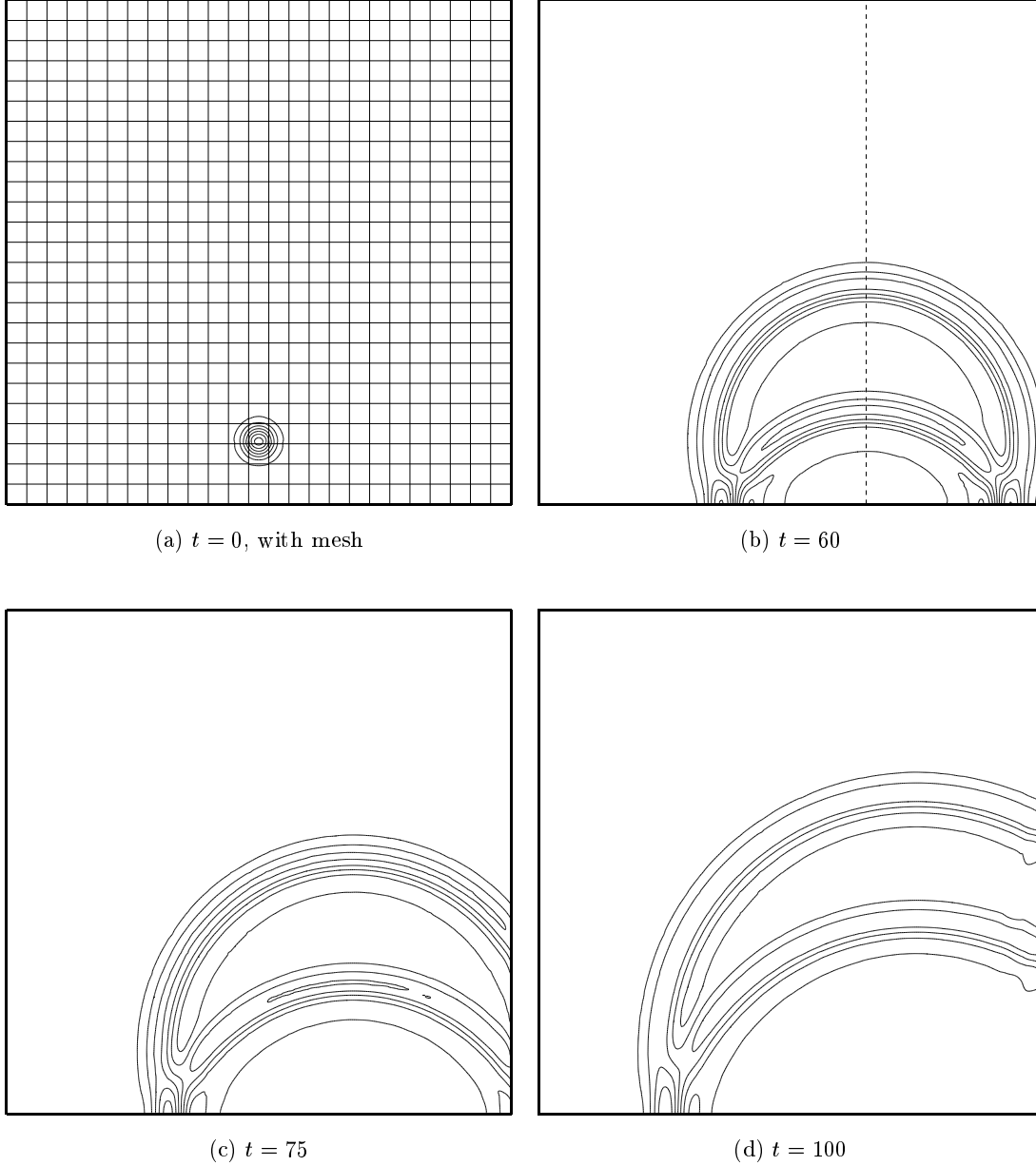


Figure 7. Gaussian-Pressure Pulse over Wall. DG(3) results, 25×25 mesh, $\nu = 0.8$. On the $t = 60$ plot, the dashed line indicates a cut along which comparisons will be made.

⁴In 2-D, DG(3) uses $\sqrt{10}$ more degrees-of-freedom per mesh direction than a conventional difference method. By this measure, DG(3) on a 25×25 mesh is equivalent to a conventional method on a 79×79 mesh.

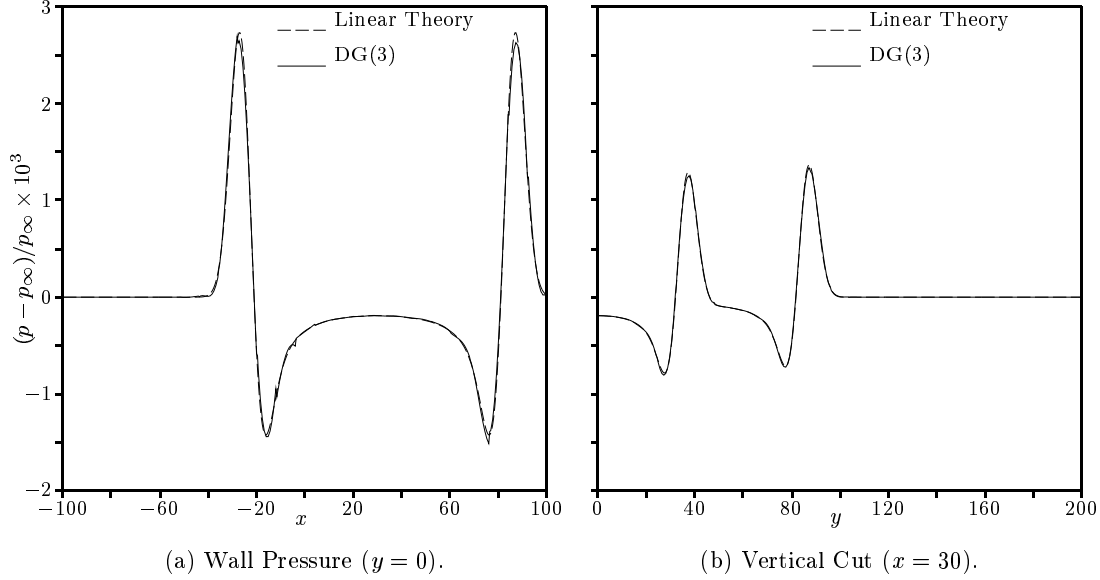


Figure 8. Gaussian-Pressure Pulse over Wall, Comparison of DG(3) with Linear Theory, Mesh 25×25 . See Figure 7b for location of vertical cut.

cells. The method does a reasonable job of maintaining symmetry, but wave reflections at the downstream boundary are noticeable at $t = 100$. These reflections are a result of the characteristic-normal boundary condition, and do not disappear with mesh refinement. A more sophisticated farfield boundary condition is needed to suppress the reflections of oblique waves.

Since the waves in this problem are weak, a full Euler solution should compare well with linear theory. Figure 8 compares the pressures at $t = 60$ along the wall, and through a vertical cut at $x = 30$. The agreement is good, except at the peaks, where DG(3) under-predicts the pressure. Note that the glitches in the predicted wall pressure, most apparent in the region $-20 \leq x \leq 0$, show the mismatch at the cell boundaries of the discontinuous interpolant. On a 50×50 mesh, DG(3) and linear theory are indistinguishable (Lowrie, 1996).

6.4. PRESSURE PULSE AND VORTEX INTERACTION

This problem is the interaction of a pressure pulse and a vortex, immersed in a steady flow. The “Category III” problem in (NAS, 1994) is very similar to the problem in this section. The solution domain is $-100 \leq x, y \leq 100$, with farfield conditions specified at all four boundaries. The initial conditions are

$$\begin{aligned} p &= 1 + 0.01\gamma\mathcal{E}_p(x, y), \\ \rho &= \gamma + 0.01\gamma [\mathcal{E}_p(x, y) + 0.1\mathcal{E}_v(x, y)], \\ u &= 0.5 \cos(\alpha) + 0.0004(y - y_v), \\ v &= 0.5 \sin(\alpha) - 0.0004(x - x_v), \end{aligned}$$

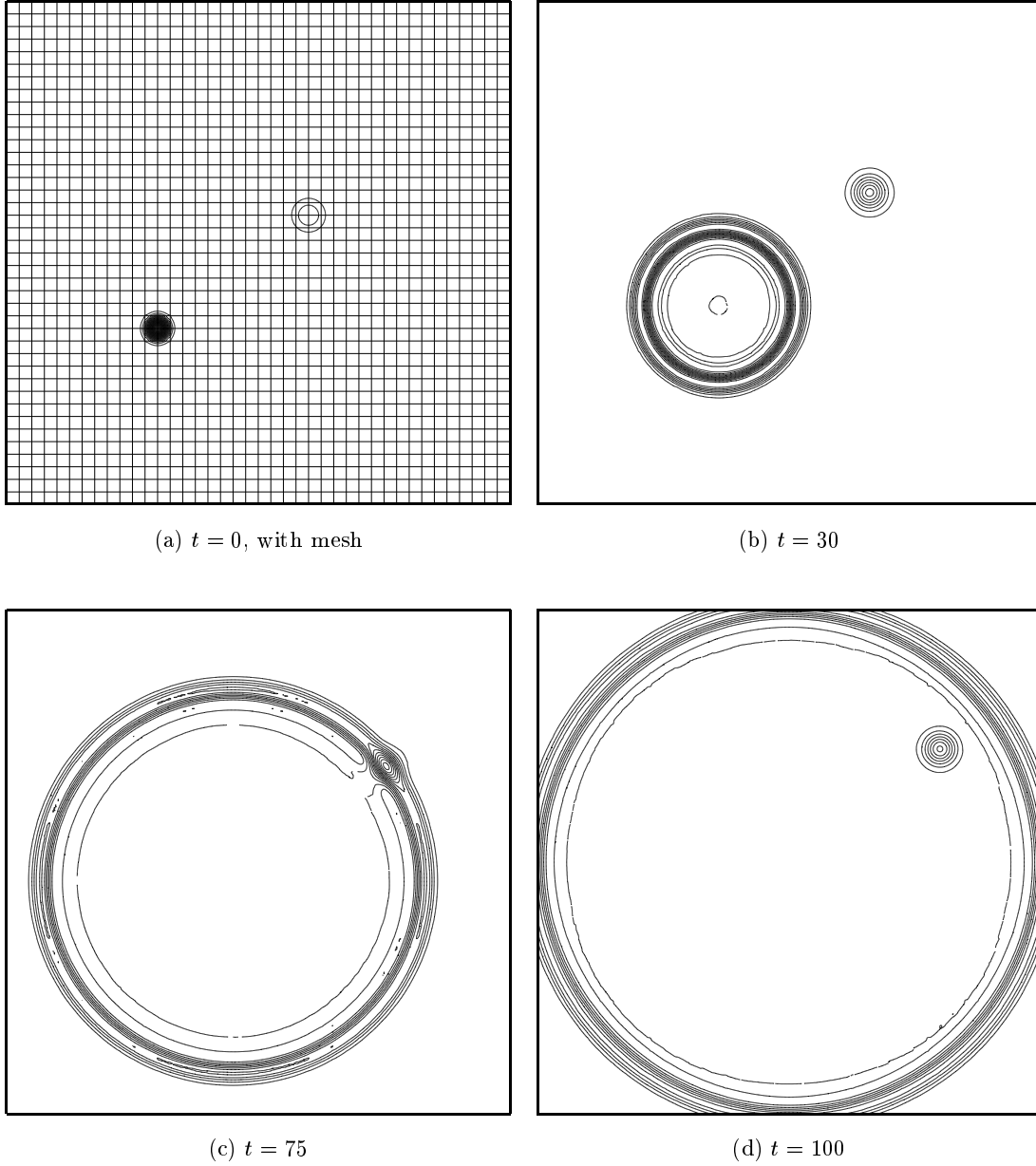


Figure 9. Pressure-Pulse / Vortex Interaction, $\alpha = \arctan(3/4)$. DG(3) Density Contours, 40×40 Mesh, $\nu = 0.8$.

where α is the freestream-flow angle, and

$$\begin{aligned}\mathcal{E}_p(x, y) &= \exp \left[-\ln(2) \frac{(x - x_p)^2 + (y - y_p)^2}{9} \right], \\ \mathcal{E}_v(x, y) &= \exp \left[-\ln(2) \frac{(x - x_v)^2 + (y - y_v)^2}{25} \right], \\ (x_p, y_p) &= -50(\cos(\alpha), \sin(\alpha)), \\ (x_v, y_v) &= 25(\cos(\alpha), \sin(\alpha)).\end{aligned}$$

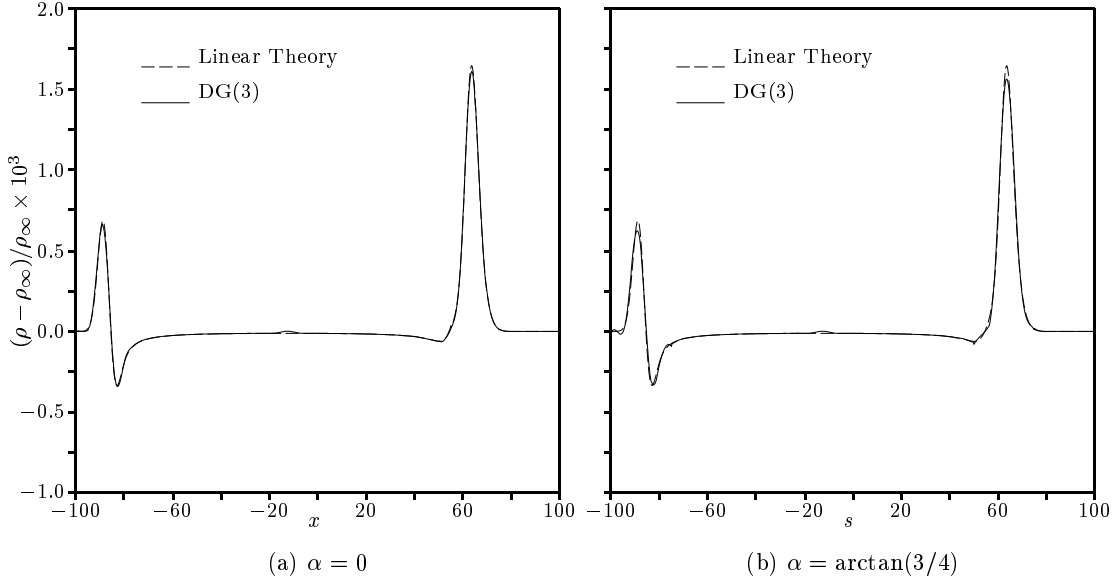


Figure 10. Pressure-Pulse / Vortex Interaction, Comparison with Linear Theory, 40×40 Mesh, $\nu = 0.8$. Cut is along line connecting pressure pulse and vortex.

Here (x_p, y_p) and (x_v, y_v) represent the initial centers of the pressure pulse and vortex, respectively. The above initial conditions differ from those in (NAS, 1994) in that the distance between the pressure pulse and vortex is independent of α .

DG(3) density contours are shown in Figure 9 for $\alpha = \arctan(3/4)$. Comparing the density contours at $t = 30$ and $t = 100$, the vortex is not distorted by the acoustic wave, as predicted by linear theory. A comparison with linear theory is made for both $\alpha = 0$ and $\alpha = \arctan(3/4)$ in Figure 10. Both solutions compare reasonably well, with those for $\alpha = \arctan(3/4)$ slightly worse. The density glitch at $x = -12.5$ corresponds to the center of the pressure pulse, and is present in the DG(k) results because the initial condition is not isentropic. Results for DG(3) on a 60×60 mesh are within 2% of linear theory (Lowrie, 1996).

7. Summary

A time-accurate method has been developed that is based on the Discontinuous Galerkin method. In deriving the method, the idea of *compactness* has been strictly followed. That is, that the discrete domain of dependence should contain a minimum amount of data outside of the physical domain of dependence. For any order-of-accuracy, the method is stable for Courant numbers less than 1, satisfies an entropy condition, and a minimization property.

We've given an argument as to why the superconvergence property is important for problems that require long-time integration. For the long-time integration of linear problems, the error is driven by a norm that converges as $\mathcal{O}(h^{2k+1})$. Although not shown here, numerical experiments indicate that this property extends to some nonlinear cases.

The use of the Staggered Mesh allows for the elimination of the Riemann problem. The

disadvantage of *any* time-staggered approach is the presence of diffusion at low Courant numbers. Although seemingly avoidable in 1-D problems, in 2-D diffusion effects are unavoidable, resulting in ‘cross-diffusion;’ see (Lowrie, 1996) for more discussion.

A comparison of $DG(k)$ with the multi-stage version (Cockburn *et al.*, 1989; Atkins and Shu, 1996) shows two main advantages in keeping the method compact. The first advantage is that the Courant restriction for the multi-stage version is inversely proportional to k , at least for $k = 1, 2$. Secondly, $DG(k)$ has a superconvergence property, although there is some evidence that increasing the order of the multi-stage integration may give the same benefit, along with a modest increase in stability. In terms of simplicity, the multi-stage version clearly shows an advantage.

The accuracy of $DG(k)$ is impressive. The typical arguments against moving beyond second-order accuracy are that higher-order methods are less robust, as a result of stability restrictions, limiting, and boundary procedures. Although much work is still needed in developing a limiter, the method described here shows that by strictly following the physics, many desirable properties can be obtained including any order-of-accuracy. The $DG(k)$ method actually improves with increasing k , in terms of the cost for a specified error tolerance. This work should open the door to the development of more practical methods that are ideal for problems that demand high accuracy.

References

- H. Atkins and C. W. Shu. A quadrature-free implementation of the discontinuous Galerkin method for hyperbolic problems. AIAA Paper 96-1683, 1996.
- B. Cockburn and C. Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws II: General framework. *Mathematics of Computation*, 52:411–435, 1989.
- B. Cockburn, S. Lin, and C. Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: One-dimensional systems. *Journal of Computational Physics*, 84:90–113, 1989.
- H. T. Huynh. Accurate upwind methods for the Euler equations. AIAA Paper 95-1737, 1995.
- J. Jaffre, C. Johnson, and A. Szepessy. Convergence of the discontinuous Galerkin finite element method for hyperbolic conservation laws. *Mathematical Models and Methods in Applied Sciences*, 5:367–386, 1995.
- C. Johnson and J. Pitkaranta. An analysis of the discontinuous Galerkin method for a scalar hyperbolic equation. *Mathematics of Computation*, 46:1–26, 1986.
- R. B. Lowrie. *Compact Higher-Order Numerical Methods for Hyperbolic Conservation Laws*. PhD thesis, University of Michigan, 1996. Available at <http://www.engin.umich.edu/research/cfd/research/publications/>.
- NASA Conference Publication 3300. *ICASE/LaRC Workshop on Benchmark Problems in Computational Aeroacoustics*, 1994.
- G. R. Richter. An optimal-order error estimate for the discontinuous Galerkin method. *Mathematics of Computation*, 50:75–88, 1988.
- P. L. Roe. Approximate Riemann solvers, parameter vectors and difference schemes. *Journal of Computational Physics*, 43, 1981.
- C. W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock capturing schemes, II. *Journal of Computational Physics*, 83:32–78, 1989.
- B. van Leer. Towards the ultimate conservative difference scheme. IV. A new approach to numerical convection. *Journal of Computational Physics*, 23, 1977.